IN THE CLAIMS:

Please amend the claims as follows:

1.   (Currently Amended)   A computer-implemented method for debugging code using a debugger comprising one or more interfaces for specifying debugger operations and displaying debug information, the method comprising:

at a halted position in the code:

accessing, via the debugger, a debug history repository comprising a plurality of history records each containing data describing code state information for a previously-encountered given state of the code;

determining whether a current state of the code matches a previously-encountered given state described in a history record in the debug history repository; and

providing an indication of the match to a user via the one or more interfaces whereby the user is allowed to view debug information contained in the history record corresponding to the match, the debug information describing at least an aspect of the previously-encountered given state, wherein the given state of the code is defined at least in part by an evaluation of a variable by a command which is separate from the code, the evaluation having been performed at the request of the user at the halted position.

2.   (Original)   The method of claim 1, wherein the given state of the code is one at which the code was halted following execution.

3.   (Original)   The method of claim 1, wherein providing the user the indication of the match comprises displaying a selectable graphical element which, when selected by the user through an input device, reveals the debug information contained in the history record corresponding to the match.

4.   (Original)   The method of claim 1, wherein the debugger and debug history repository are part of a distributed environment.

5.   (Original)   The method of claim 1, wherein the code is selected from the group consisting of procedural program code, object oriented program code, and combinations thereof.

6.   (Original)   The method of claim 1, wherein the given state of the code is defined at least in part by a debugger control point encountered during execution of the code and having caused the code to be stopped by the debugger at the halted position.

7.   (Cancelled)   The method of claim 1, wherein the given state of the code is defined at least in part by an evaluation of a variable, the evaluation having been performed at the request of the user at the halted position.

8.   (Currently Amended)   The method of claim [[7]] 1, wherein providing the user the indication of the match comprises displaying a series of previously performed variable evaluations by commands which are separate from the code performed during a previous stoppage of the code at the halted position.

9.   (Currently Amended)   A computer-implemented method for debugging code using a debugger comprising one or more interfaces for specifying debugger operations and displaying debug information, the method comprising:
       at a halted position in the code:
              accessing, via the debugger, a debug history repository comprising a plurality of history records each containing (i) data describing code state information for a given state of the code while stopped at a halted position and (ii) data describing a series of variable evaluations by commands which are separate from the code performed at the request of a user during a stoppage at a position in the code;
              performing at least one evaluation of one or more variables in the code by a command which is separate from the code at the request of the user; and
              for each evaluation: (i) determining whether a current state of the code matches a given state described in a history record in the debug history

repository; and (ii) if so, providing a user an indication of the match via the one or more interfaces.

10.   (Original)   The method of claim 9, wherein providing the user the indication of the match comprises displaying a selectable graphical element which, when selected by the user through an input device, reveals the debug information contained in the history record corresponding to the match.

11.   (Original)   The method of claim 9, wherein providing the user the indication of the match comprises displaying the data describing the series of variable evaluations contained in the history record corresponding to the match.

12.   (Original)   The method of claim 9, wherein providing the user the indication of the match comprises displaying a selectable graphical element which, when selected by the user through an input device, reveals debug information contained in the history record corresponding to the match.

13.   (Currently Amended)   The method of claim 12, wherein the debug information comprises user commentary entered by the user as text using a text box.

14.   (Currently Amended)   The method of claim 12, wherein the debug information comprises user commentary entered by the user as text using a text box describing debugging operations performed during a previous stoppage of the code while in a state matching the current state.

15.   (Original)   The method of claim 9, further comprising, after performing the at least one evaluation, displaying a user interface configured to allow the user to create a history record for the at least one evaluation.

16.   (Original)   The method of claim 15, wherein the user interface is configured to allow the user to specify a state to be associated with the history record to be created.

17.   (Currently Amended)   The method of claim 15, wherein the at least one evaluation comprises a plurality of evaluations by commands which are separate from

the code performed at the request of a user and wherein the user interface displays the plurality of evaluations as selectable items from which the user is allowed to select in order to designate one of the plurality of evaluations as a ~~primary~~ main state for the history record to be created, wherein the ~~primary~~ main state is used to determine a match between the history record to be created and a subsequent state at a subsequent halted position in the code.

18.   (Currently Amended)   A computer readable storage medium containing a debugger program which, when executed, performs an operation to facilitate debugging of code, the operation comprising:

at a current halted position, at which execution of the code is suspended for the purpose of allowing a user to examine the code:

collecting state information related to the current halted position wherein the state information is defined at least in part by an evaluation of one or more variables by a command which is separate from the code, the evaluation having been performed at the request of a user at the halted position;

querying a debug history repository to determine whether a current state of the code at the current halted position matches a given state described in a history record in the debug history repository; and

if so, providing a user an indication of the match via the one or more interfaces.

19.   (Currently Amended)   The computer readable storage medium of claim 18, wherein the debugger program allows a user to enter a new record in the debug history repository if a matching record is not found.

20.   (Currently Amended)   The computer readable storage medium of claim 18, wherein the debugger program automatically creates a new record in the debug history repository if a matching record is not found.

21. (Cancelled) The computer readable medium of claim 18, wherein the state information is defined at least in part by the evaluation of one or more variables, the evaluation having been performed at the request of the user at the halted position.

22. (Currently Amended) The computer readable storage medium of claim 18, wherein the debug history repository is queried using filters selected from the group consisting of developer name, time, and combinations thereof.

23. (Currently Amended) A computer system for debugging code, comprising:

a computer readable storage medium including:

a debug history repository comprising a plurality of history records each containing data describing code state information for a given state of the code under debug wherein each history record further contains data describing a series of variable evaluations by commands which are separate from the code performed at the request of a user during a stoppage at a position in the code; and

instructions for a debugger comprising one or more interfaces for specifying debugger operations and displaying debug information; wherein the debugger is configured to:

a processor, which, when executing the instructions for the debugger, is configured to:

create the plurality of history records in the debug history repository;

access the plurality of history records in the debug history repository;

determine, at a given halted position in the code, whether a current state of the code matches a given state described in a history record of the plurality of history records; and

if a matching state is determined, provide a user an indication of the match via the one or more interfaces while at the given halted position, whereby the user is allowed to view debug information contained in the history record corresponding to the match.

543296_1.DOC

24.   (Currently Amended)   The <u>computer</u> system of claim 23, wherein the repository is a database selected from the group consisting of a relational database, object-relational database, extensible-markup-language (XML) database, and combinations thereof.

25.   (Currently Amended)   The <u>computer</u> system of claim 23, wherein the <u>processor, when executing the instructions for the</u> debugger, <u>is configured to query</u> ~~queries~~ the debug history repository based on available current state information.

26.   (Currently Amended)   The <u>computer</u> system of claim 23, wherein the one or more interfaces comprise a graphical user interface.

27.   (Currently Amended)   The <u>computer</u> system of claim 23, wherein <u>the</u> <u>processor, when executing the instructions for</u> the debugger<u>,</u> is configured to provide the user the indication of the match by displaying a selectable graphical element which, when selected by the user through an input device, reveals debug information contained in the history record corresponding to the match.

28.   (Currently Amended)   The <u>computer</u> system of claim 27, wherein the debug information comprises user commentary<u> entered by the user as text using a text box</u>.

29.   (Currently Amended)   The <u>computer</u> system of claim 27, wherein the debug information comprises user commentary <u>entered by the user as text using a text box</u> describing debugging operations performed during a previous stoppage of the code while in a state matching as the current state.

30.   (Cancelled)   The system of claim 23, wherein each history record further contains data describing a series of variable evaluations performed during a stoppage at a position in the code.

31.   (Currently Amended)   The <u>computer</u> system of claim [[30]] <u>23</u>, wherein the <u>processor, when executing the instructions for the</u> debugger<u>,</u> is further configured to perform at least one evaluation of one or more variables in the code <u>by commands</u> <u>which are separate from the code at the request of the user</u> at the halted position; and

wherein the <u>processor, when executing the instructions for the</u> debugger<u>,</u> performs the determination of whether the current state of the code matches a given state described in a history record for each evaluation; and wherein, if the matching state is determined and the indication of the match provided, the user is allowed to view the data contained in the history record corresponding to the match and describing the series of variable evaluations performed during the stoppage at a position in the code.

32.   (Currently Amended)   The <u>computer</u> system of claim [[30]] <u>23</u>, wherein the history record corresponding to the match further comprises user commentary<u> entered by the user as text using a text box</u>.

33.   (Currently Amended)   The <u>computer</u> system of claim [[30]] <u>23</u>, wherein the history record corresponding to the match further comprises user commentary <u>entered by the user as text using a text box</u> describing debugging operations performed during a previous stoppage of the code while in a state matching the current state.

34.   (Currently Amended)   The <u>computer</u> system of claim [[30]] <u>23</u>, wherein the <u>processor, when executing the instructions for the</u> debugger<u>,</u> is configured to display a user interface configured to allow the user to create a history record for the at least one evaluation.

35.   (Currently Amended)   The <u>computer</u> system of claim 34, wherein the user interface is configured to allow the user to specify a state to be associated with the history record to be created.

36.   (Currently Amended)   The <u>computer</u> system of claim 34, wherein the at least one evaluation comprises a plurality of evaluations <u>by commands which are separate from the code</u> and wherein the user interface displays the plurality of evaluations as selectable items from which the user is allowed to select in order to designate one of the plurality of evaluations as a ~~primary~~ <u>main</u> state for the history record to be created, wherein the ~~primary~~ <u>main</u> state is used to determine a match between a state described by code state information contained in the history record to be created and a subsequent state at a subsequent halted position in the code.

37.   (Currently Amended)   The <u>computer</u> system of claim 34, wherein the at least one evaluation comprises a plurality of evaluations and wherein the user interface displays the plurality of evaluations as selectable items from which the user is allowed to select in order to designate which of the plurality of evaluations are to be included in the history record to be created.